

REPUBLIQUE DU CAMEROUN

.....
Paix -Travail-Patrie
.....

**MINISTERE DE L'ENSEIGNEMENT
SUPERIEUR**



REPUBLIC OF CAMEROON

.....
Peace- Work –Fatherland
.....

MINISTRY OF HIGHER EDUCATION



PROJET TUTORÉ

**Thème : Mise en place d'un système de détection
d'intrusions utilisant SNORT**

Rédigé et Soutenu par :

TIOWA NZONTEU

&

YEMBA STEVE

En vue de l'obtention du :

Diplôme de Master des Systèmes d'Information et d'Infrastructure

Sous la direction de :

Encadreur professionnel

Mr LONTSI Leonel

Encadreur académique

Mr TSOBENG David

Année académique : 2021 -2022

Dédicace
A
Nos Familles

REMERCIEMENTS :

Nous tenons à adresser toute ma gratitude et mes sincères remerciements aux personnes qui ont contribué à la réussite de notre formation :

- Nos familles pour leurs prières, soutien et sacrifices
- Monsieur NGUIMEZAP Paul pour avoir mis sur pied le pôle de l'excellence académique
- Madame NOUBANKA Manuella à la tête du département 3IL CS2I
- Monsieur LONTSI Leonel Notre encadreur professionnel
- Monsieur TSOBENG DAVID Notre encadreur académique
- Tous nos enseignants pour les connaissances acquises
- Tous nos amis pour l'aide perçus

Et tous ceux qui n'ont pas pu être mentionné plus haut.

SOMMAIRE

INTRODUCTION GÉNÉRALE	2
I. CONTEXTE ET JUSTIFICATION	2
II. OBJECTIFS	3
CHAPITRE I : ETAT DE L'ART	5
I. INTRUSION.....	5
II. DETECTION ET PREVENTION D'INTRUSIONS	5
III. HISTORIQUE	6
IV. ARCHITECTURE INTERNE D'UN IDS.....	7
CHAPITRE II : PLAN ET MÉTHODOLOGIE DU PROJET	18
I. PLAN DU PROJET	18
II. METHODOLOGIE.....	19
CHAPITRE III INSTALLATION ET CONFIGURATION	28
I. ESTIMATION DE LA SOLUTION	28
II. INSTALLATION DES PREREQUIS	28
III. INSTALLATION DES DEPENDANCES	29
IV. CREATION DE REGLES.....	39
CONCLUSION GÉNÉRALE	42
BIBLIOGRAPHIE ET WEBOGRAPHIE	43
I. BIBLIOGRAPHIE	43
II. WEBOGRAPHIE	43

Liste des figures

Figure 1 Architecture d'un système de détection d'intrusions selon l'IDWG	6
Figure 2 Déploiement typique d'un NIDS	15
Figure 3 Autre déploiement de NIDS	16
Figure 4 Diagramme de Gantt du projet.....	18
Figure 5 Choix du placement d'un IDS	20
Figure 6 Architecture de SNORT.....	24
Figure 7 Le décodeur de paquets.....	25
Figure 8 Choisir une interface réseau par défaut.....	30
Figure 9 Choisir le mode de démarrage de SNORT.	31
Figure 10 Récupération d'adresse ip par défaut.....	31
Figure 11 Le paramétrage du mode.....	32
Figure 12 Suggestion d'ajout d'autres options pour SNORT.	32
Figure 13 Le choix de recevoir des alertes par e-mail.	33
Figure 14 Choisir une interface par défaut.....	34
Figure 15 Entrer l'adresse ip du serveur.	34
Figure 16 Paramétrage du mode.....	35
Figure 17 Le choix de recevoir des alertes par e-mail.	35
Figure 18 Vérification de la base de données.....	36
Figure 19 Donner un nom à la machine serveur.	36
Figure 20 Donner un nom à la base de données.....	37
Figure 21 Donner un nom à l'utilisateur.	37
Figure 22 Accorder un mot de passe lors de connexion.....	38
Figure 23 La fin de configuration du mysql.....	38
Figure 24 Remplissage du fichier 'local.rules' édité.....	39
Figure 25 Création de la base de données.	39
Figure 26 Activation de la base de données.....	40

Liste des tableaux

Tableau 1 Plan du projet.....	18
-------------------------------	----

GLOSSAIRE :

SNORT: The Open Source Network Intrusion Detection System.

DMZ: Demilitarized Zone.

IDS: Intrusion Detection System.

IPS: Intrusion Prevention System.

VPN: Virtual Private Network.

DNS: Domain Name Service.

DOS: Denial of Service.

NIDS: Network Intrusion Detection System.

NIPS: Network Intrusion Prevention System.

HIDS: Host based Intrusion Detection System.

HIPS: Host based Intrusion Prevention System.

HTML: HyperText Markup Language.

HTTP: HyperText Transfer Protocol.

IDS: Intrusion Detection System.

INTERNET: Interconnected Network.

IP: Internet Protocol.

IPS: Intrusion Prevention System.

ISO: International Standards Organization.

LAN: Local Area Network.

MySQL: My Structured Query Language.

PDU: Protocol Data Unit.

TCP: Transmission Control Protocol.

TELNET: Terminal Network.

FTP: File Transfer Protocol.

UDP: User Datagram Protocol.

WWW: World Wide Web.

RESUME :

La sécurité de nos jours est un problème d'une importance capitale, elle est devenue un problème majeur dans la gestion des réseaux d'entreprises ainsi que pour les particuliers. Différents mécanismes ont été mis en place pour faire face à ces problèmes de sécurité, comme les antivirus, les pare-feux, le cryptage, mais ces mécanismes ont des limites face au développement rapide des techniques de piratage. Pour éviter ces limites, l'utilisation des systèmes de détection d'intrusion s'impose.

Les systèmes de détection d'intrusions ont été conçus pour une surveillance continue, et la découverte des violations de la politique de sécurité, ainsi l'identification de toute activité non autorisée dans un réseau.

C'est dans cette optique que s'inscrit notre projet tutoré. À savoir l'étude d'un système de détection d'intrusions utilisant SNORT, et sa mise en place pour sécuriser un réseau informatique.

Mots clés : réseau, antivirus, pare-feu, cryptage, piratage, système de détection d'intrusion, politique de sécurité, SNORT, réseau informatique

ABSTRACT :

Security nowadays is a problem of paramount importance, it has become a major problem in the management of business networks as well as for individuals. Different mechanisms have been put in place to deal with these security problems, such as antivirus, firewalls, encryption, but these mechanisms have limits in the face of the rapid development of hacking techniques. To avoid these limits, the use of intrusion detection systems is essential.

Intrusion detection systems are designed for continuous monitoring and discovery of security policy violations, thus identifying any unauthorized activity in a network.

It is in this perspective that our tutored project fits. Namely the study of a system of

Intrusion detection using SNORT, and its implementation to secure a computer network.

Keywords: network, antivirus, firewall, encryption, hacking, intrusion detection system, security policy, SNORT, computer network

Chapitre 0

Introduction Générale

Introduction générale

Les réseaux informatiques sont devenus des ressources vitales et déterministes pour le bon fonctionnement des entreprises. De plus, ces réseaux sont ouverts de fait qu'ils sont pour la plus parts raccordés à l'Internet.

Cette ouverture qui permet de faciliter la communication, engendre malheureusement des risques importants dans le domaine de la sécurité informatique. Les utilisateurs de l'Internet ne sont pas forcement pleins de bonnes intentions, ils peuvent exploiter les vulnérabilités des réseaux et systèmes pour réaliser leurs attaques. Les conséquences de ces attaques peuvent être lourdes pour un particulier (pertes d'informations, ou pire encore vol d'informations, atteinte à la vie privée...) et pour une entreprise (perte du savoir-faire, atteinte à l'image de marque, perte financière...). Pour cela, les administrateurs déploient des solutions de sécurité efficace capable de protéger le réseau de l'entreprise. Dans ce contexte, les IDS constituent une bonne alternative pour mieux protéger le réseau informatique.

Un système de détection d'intrusion (IDS) est un mécanisme écoutant le trafic réseau de manière furtive afin de repérer des activités anormales ou suspectes et permettant aussi d'avoir une action de prévention sur les risques d'intrusion.

Dans le cadre de ce projet nous nous intéresserons aux outils de détection d'intrusions réseaux (IDS) plus particulièrement à SNORT, permettant de détecter des intrusions réseau à temps réel.

I. Contexte et Justification

L'augmentation des attaques et la complexité de celles-ci compliquent de plus en plus la mise en place et la gestion des réseaux. En effet, il est nécessaire de garantir la disponibilité de chaque équipement du réseau. De la même manière l'intégrité et la confidentialité des données constituent un enjeu important. Les attaques informatiques peuvent se retrouver sous plusieurs formes à savoir virus, vers, chevaux de Troie et autres. Cette variété déjoue presque toujours, tôt ou tard, les moyens de sécurité, justifiant largement la locution "le risque zéro n'existe pas". De nouveaux outils sont inventés, de nouveaux concepts apparaissent, de nouvelles politiques

de sécurité sont implémentées sans pour autant offrir une sécurité entière aux systèmes d'information. Ainsi, malgré toutes les stratégies de sécurité, les menaces n'ont jamais cessé d'exister. Tout réseau ouvert à d'autres réseaux est alors sous menace permanente. Dans le contexte, un IDS particulièrement SNORT qui sera utilisé tout au long de notre travail constituent une bonne alternative pour mieux protéger le réseau informatique.

II. Objectifs

1. Objectif principal

L'objectif principal qui nous guide dans ce travail est de mettre en place un système de détection d'intrusion plus particulièrement à SNORT, permettant de détecter des intrusions à temps réel dans le réseau de la structure permettant ainsi d'augmenter le niveau de sécurité des services ainsi accroître la QoS de l'entreprise.

2. Objectifs spécifiques

- Etudier et analyser tous les aspects traités par un système de détection / prévention d'intrusion réseau.
- Etude de cas : SNORT
- Installation et configuration de SNORT
- Tests de détection d'intrusion en utilisant des règles prédéfinies de SNORT.
- Tests et évaluations de performance de IDS SNORT.

Ce rapport va être organisé comme suit :

Dans le premier chapitre, nous présentons l'état de l'art. Dans le second chapitre, nous allons détailler les matériels et méthodes utilisés. Enfin, le dernier chapitre sera constitué à l'installation et à la configuration de l'outil de détection intrusion SNORT.

Chapitre I

Etat de l'art

Chapitre I : Etat de l'art

Les systèmes de détection et de prévention d'intrusions sont utilisés dans un réseau pour détecter les attaques externes ou internes. Dans ce chapitre, nous présentons une analyse détaillée de ces systèmes, en mettant en évidence leurs forces et faiblesses pour la protection des systèmes informatiques.

I. Intrusion

De façon générale, une intrusion est un accès non autorisé à une ressource, une société, un groupe ou un système d'information. En informatique, elle désigne toute activité qui viole la politique de sécurité d'un système ou qui essaie de prendre en défaut le mécanisme de sécurité d'une organisation. C'est toute tentative réussie ou non d'exploitation de vulnérabilités, de failles de sécurité. Elle peut être exécutée depuis l'intérieur du réseau ou par des individus situés à l'extérieur et qui tentent de passer au travers des mécanismes de sécurité mis en place.

II. Détection et prévention d'intrusions

Un système de détection d'intrusions (ou IDS : Intrusion Detection System) est un outil logiciel destiné à repérer des activités anormales ou suspectes sur un système (un réseau ou un hôte).

C'est un outil qui essaie d'identifier toute introduction illégale ou tout comportement anormal sur un système d'information. Il est un ensemble de composants logiciels et matériels dont la fonction principale est de détecter et d'analyser toute tentative d'effraction au sein du système.

Lorsque la détection est suivie de solutions actives, alors on parle de système de prévention d'intrusions (IPS pour Intrusion Prevention System). Le principal avantage des IDS/IPS par rapport à tout autre système tel que les pare-feux est leur capacité à accéder aux contenus même des paquets et les analyser. La détection ne porte donc plus seulement sur les en-têtes de protocoles comme c'est le cas pour les pare-feux. Dans la suite de ce développement, le thème IDS sera utilisé pour se référer aux deux catégories et toute distinction particulière sera clairement exposée.

III. Historique

La détection d'intrusions tire ses origines des systèmes d'audit. L'objectif était d'automatiser l'audit des systèmes que jusqu'alors était fait manuellement par les administrateurs réseaux. Il s'agit bien, théoriquement, de détecter de manière automatique les violations de politique de sécurité ou de droit, qu'on appelle intrusions. Les premiers systèmes de détection d'intrusions ont été initiés par l'armée américaine qui publia dans les années 1970 les objectifs d'un système de sécurité, parmi lesquels figure la détection de toute tentative de violation de mécanisme de protection. Les premiers travaux ont réellement débuté avec J.P. Anderson en 1980 qui décrit dans une publication comment améliorer les mécanismes de sécurité. Se servant de ces travaux, Dorothy Denning et Peter Neumann proposèrent en 1987 un modèle théorique d'un système de détection d'intrusions. Vers la fin des années 1980, Todd Heberlein introduit l'idée de la détection d'intrusion réseau. Il développa en 1990 le NSM (Network Security Monitor) qui était le premier système de détection d'intrusions réseau. En 1992, U.S. Air Force, UC Davis et d'autres ont développé le concept du système de détection d'intrusion distribué (DIDS pour Distributed Intrusion Detection System), puis ont introduit l'approche hybride de la détection d'intrusions.

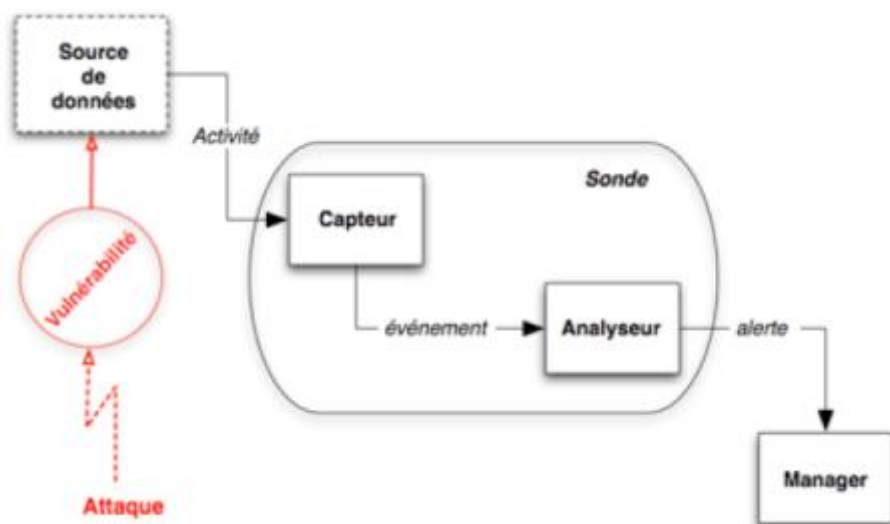


Figure 1 Architecture d'un système de détection d'intrusions selon l'IDWG

IV. Architecture interne d'un IDS

L'architecture d'un système de détection d'intrusions dispose de trois composants communs à la majorité des IDS. Selon le modèle proposé par IDWG (Intrusion Detection Working Group) Le capteur reçoit les données sources brutes c'est-à-dire les paquets réseau et les données d'audit. Il envoie ensuite des événements à l'endroit de l'analyste. Ce dernier vérifie si certains événements sont caractéristiques d'activités malveillantes, auquel cas il génère des alertes qu'il envoie au manager. Ce dernier se charge de présenter les alertes à l'opérateur puis décide éventuellement de la réaction à adopter.

1. Le capteur

Le capteur est l'outil utilisé pour enregistrer les données brutes. Il est responsable de la collecte des données depuis le système surveillé. C'est le premier composant de la chaîne de détection à entrer en activité. Il permet donc de disposer des informations à analyser. Un pré-traitement peut être effectué sur les données à ce niveau notamment le filtrage pour éliminer les données non pertinentes ; ce qui permet de réduire la quantité de données à analyser par la suite. Il présente ensuite ces données sous forme d'événements à l'analyste.

On distingue trois types de capteurs en fonction des sources de données qu'ils utilisent :

- Les capteurs systèmes qui utilisent les données provenant des journaux d'audit des systèmes d'exploitation des machines et des appels systèmes effectués par les applications ;
- Les capteurs réseau qui écoutent les communications entre les différentes machines du réseau à travers une interface spécifique ;
- Les capteurs applicatifs qui se chargent d'enregistrer les données produites par les applications elles-mêmes. Ceci permet de suivre le fonctionnement d'une application spécifique.

2. L'analyste

L'analyste a pour objectif principal d'analyser le flux d'événements envoyé par le capteur pour identifier des données caractéristiques d'activités malveillantes. Il utilise alors ces

événements afin de déceler une possible intrusion et génère en conséquence des alertes. Ces alertes sont ensuite présentées au manager.

3. Le manager

Le manager a pour rôle de traiter les alertes fournies par l'analyseur puis de les présenter à l'administrateur. Une corrélation d'alertes est nécessaire dans le cas de plusieurs sources d'alerte. Il peut également déclencher des mesures de traitement comme les actions suivantes :

- confinement de l'attaque qui a pour but de limiter les effets possibles ;
- éradication de l'attaque qui tente de l'arrêter ;
- recouvrement qui est l'étape de restauration du système dans un état sain ;
- diagnostic qui est la phase d'identification du problème, de ses causes et qui peut éventuellement être suivi d'actions contre l'attaquant (fonction de réaction).

Du fait du manque de fiabilité des systèmes de détection d'intrusions actuels, les réactions sont rarement automatisées, car elles peuvent se traduire par un déni de service en cas de réaction à des faux positifs.

4. Terminologie relative aux systèmes de détection d'intrusions

a. Faux Positif

Fausse alerte levée par le système de détection d'intrusions. C'est une alerte provenant d'un IDS et qui ne correspond pas à une attaque réelle. L'idéal est de ne pas avoir ce type d'alarme. Comme conséquence néfaste, lorsque l'administrateur de sécurité est inondé par ce type d'alerte, cela constitue une source d'ennui et l'amène parfois à ignorer des alertes réelles. De plus, cela peut entraîner une paralysie du réseau surtout dans le cas des IPS.

b. Vrai positif

Cela se réfère à une alarme où le système de détection d'intrusion reporte une intrusion réelle. Cela signifie qu'un système est entrain d'être compromis. Idéalement toutes les attaques devraient être détectées.

c. Faux négatif

Attaque non repérée par le système de détection d'intrusions. C'est une intrusion réelle qui n'a pas été détectée. Ceci est un comportement indésirable du système de détection d'intrusion pouvant avoir plusieurs origines. Un IDS mal positionné privé d'une partie du trafic qu'il devrait analyser peut omettre des attaques. Aussi, cela pourrait être le cas lorsqu'il ne peut supporter le taux de trafic du réseau ; des paquets sont donc perdus.

d. Vrai négatif

Ce terme est utilisé pour décrire le cas où le système de détection d'intrusions trouve qu'un paquet est inoffensif et que cela est vrai.

e. Evasion

C'est une technique utilisée pour dissimuler une attaque et faire en sorte qu'elle ne soit pas décelée par le Système de détection d'intrusions.

f. Sonde

C'est l'élément de l'architecture IDS qui collecte les informations brutes et en fournit une description à l'aide d'événements. C'est un ou plusieurs capteurs couplés avec un analyseur.

5. Les types de système de détection d'intrusions

Les données utilisées par les systèmes de détection d'intrusions peuvent provenir de sources variées. Elles peuvent être recueillies sur une machine ou sur tout un réseau. Il existe donc différents types d'IDS que nous allons décrire dans cette section.

a. Les systèmes de détection d'intrusions de type hôte (HIDS)

Un système de détection d'intrusions de type hôte (HIDS pour Host-based Intrusion Detection System) est caractérisé par l'analyse des événements et traces générés par le système d'une machine. Son objectif est de surveiller l'activité d'une machine unique donnée. A travers sa sonde logée sur un hôte singulier, il collecte des données produites par les systèmes d'exploitation des machines, notamment par le biais des journaux d'audit système ou par celui des appels système invoqués par les applications. Il s'insère entre les applications et le cœur du système d'exploitation pour protéger des applications ou des serveurs critiques. Son utilisation dans un réseau nécessite son installation sur chacun des systèmes à sécuriser. Dans des

environnements plus larges, son déploiement est prohibitif en termes de coût et de maintenance. Son principal avantage est sa vue beaucoup plus profonde sur l'activité du système et se révèle plus précis dans sa stratégie de détection. Dans cette catégorie, peuvent être distingués les IDS de niveau application, les programmes de vérification d'intégrité de fichiers et les IDS de niveau système. Les premiers examinent les opérations apparues dans une application pour détecter si elle est manipulée ou non ou si elle n'effectue pas des accès interdits vers des données sensibles.

Une fois qu'un pirate prend le contrôle d'une machine par une application, il va essayer d'injecter du code malicieux. Cela pourrait affecter le système de fichiers. Il modifie le comportement de certains fichiers critiques tels que les dll (dynamically linked libraries), des programmes afin de créer une porte dérobée ou de rester indétectable tout en perpétrant son attaque. Ainsi certains programmes tels que Tripwire 3 sont utilisés pour vérifier l'intégrité des fichiers importants. Enfin certains IDS sont beaucoup rattachés au noyau système. Un HIDS a une vue totalement limitée à un hôte, donc incapable de détecter les attaques ciblant plusieurs machines. Alors les IDS réseau font leur apparition.

b. Les systèmes de détection d'intrusions de type réseau (NIDS)

Un système de détection d'intrusions de type réseau (NIDS pour Network Intrusion Detection System) utilise les données réseau. Il analyse les paquets transitant sur un réseau afin d'identifier des anomalies. Son objectif est d'inspecter le trafic d'un grand nombre d'hôtes. Il écoute donc tout le trafic réseau. Trois technologies peuvent être distinguées à savoir :

Promiscuous-mode NIDS : ils fonctionnent en mettant une interface en mode promiscuous c'est-à-dire en écoute sur le réseau. Cela leur permet de pouvoir accéder à tout échange dans le réseau surveillé.

Network Node IDS : ils s'intéressent seulement à des hôtes donnés. En effet les réseaux commutés empêchent les IDS précédents (Promiscuousmode NIDS) de bien fonctionner car les paquets ne sont plus diffusés.

Aussi la montée en débit des réseaux actuels entraîne la perte de paquets qui à priori est indésirable. Alors les Network Node IDS résolvent ces problèmes en spécialisant leurs analyses sur des hôtes particuliers.

Wireless Intrusion Detection Systems : ce sont les types d'IDS destinés aux Réseaux Wifi. Ils sont réalisés pour surveiller les réseaux utilisant ce protocole qui n'est pas conçu à priori

avec un esprit de sécurité. Kismet, NetSumbler, AirIDS, wIDS et SNORT-Wireless en sont des exemples.

Contrairement à un HIDS qui est restreint à un hôte, le NIDS à une vue plus générale sur l'ensemble du réseau ou du sous-réseau. Il permet donc de surveiller à moins de ressources tout un ensemble d'hôtes. Cette caractéristique simplifie le déploiement et la maintenance d'une solution de détection visant à garantir une couverture optimale du réseau surveillé. L'approche système est plus complexe à déployer car elle nécessite une multiplication du nombre de capteurs dans le réseau. De plus, le coût engendré par la collecte des données par ces capteurs peut dégrader sensiblement les performances des systèmes sur lesquels ils sont installés. Cependant, on peut s'interroger sur la pérennité des capteurs réseaux pour trois raisons principales. Premièrement, la montée en débit des réseaux contraint fortement les capacités de collecte de l'intégralité du trafic. Les constructeurs de NIDS ont recours à des capteurs matériels spécifiques pour accélérer la collecte, mais la détection d'intrusions dans le cœur de réseau peut poser problème car seules certaines données peuvent être prises en compte.

L'inspection de la totalité des paquets n'étant pas envisageable, les IDS pour les réseaux à haut débit doivent échantillonner les données et l'analyse ne porte souvent que sur l'entête et la détection reste imprécise. Deuxièmement, les capteurs réseau ne peuvent analyser le trafic chiffré. Or, la prise en compte progressive des problèmes de sécurité tend à généraliser l'utilisation du chiffrement dans les protocoles réseau, rendant à terme les capteurs réseau inopérants. Enfin, l'analyse seule du trafic réseau s'avère souvent insuffisante pour assurer une détection fiable et pertinente des violations de politique de sécurité, l'IDS ne disposant que de trop peu d'informations sur les systèmes attaqués. Par ailleurs les NIDS sont en général victimes d'évasion et des attaques par déni de service.

c. Les solutions hybrides

Les IDS de type hybride intègrent les deux technologies précédentes dans leur fonctionnement pour bénéficier simultanément des avantages de l'une et de l'autre. Leur objectif est d'analyser les paquets réseaux mais aussi de suivre ce qui se passe exactement au niveau d'une machine de façon individuelle.

d. Les IPS

Contrairement à l'analyse passive réalisée par un IDS, un système de prévention d'intrusions (IPS pour Intrusion Prevention System) analyse de façon active les paquets. C'est un ensemble

de composants logiciels et matériels dont la fonction principale est d'empêcher toute activité suspecte détectée au sein d'un système. En gros, il fonctionne de manière similaire à un IDS mais non seulement il détecte l'intrusion mais aussi il prend des mesures actives contre celle-ci.

Plusieurs stratégies de prévention d'intrusions existent :

- protection de mémoire et de processus (host-based memory and process protection): surveille l'exécution des processus et les tue s'ils ont l'air dangereux (buffer overflow). Cette technologie est utilisée dans les KIPS (Kernel Intrusion Prevention System).
- interception de session (session interception / session sniping) : termine une session TCP avec la commande TCP Reset : « RST ». Ceci est utilisé dans les NIPS.
- routeur détecteur d'intrusions (gateway intrusion detection) : si un système NIPS est placé en tant que routeur, il bloque le trafic ; sinon il envoie des messages à d'autres routeurs pour modifier leur liste d'accès.

Un IPS possède de nombreux inconvénients. En effet, il bloque toute activité qui lui semble suspecte. Or, il est quasiment impossible d'assurer une fiabilité complète dans l'identification des attaques. Un IPS peut donc malencontreusement bloquer du trafic inoffensif. Par exemple, un IPS peut détecter une tentative de déni de service alors qu'il s'agit d'une période chargée en trafic. Les faux positifs sont donc très dangereux pour les IPS. Un autre inconvénient est qu'un pirate peut utiliser sa fonctionnalité de blocage pour mettre hors service un système. L'exemple d'un individu mal intentionné qui attaque un système protégé par un IPS, tout en spoofant son adresse IP en est un cas. Si l'adresse IP spoofée est celle d'un nœud important du réseau, les conséquences seront catastrophiques. Pour pallier ce problème, de nombreux IPS disposent des « white lists », c'est-à-dire des listes d'adresses réseaux qu'il ne faut en aucun cas bloquer.

Autre inconvénient et non le moindre est qu'un IPS est peu discret. En effet, à chaque blocage d'attaque, il montre sa présence. Cela peut paraître anodin, mais si un pirate remarque la présence d'un IPS, il tentera de trouver une faille dans celui-ci afin de réintégrer son attaque mais cette fois en passant inaperçu. Voilà pourquoi les IDS passifs sont souvent préférés aux IPS. Cependant, il est intéressant de noter que plusieurs IDS (Ex : Bro, SNORT, RealSecure, Dragon, ...) ont été dotés d'une fonctionnalité de réaction automatique à certains types d'attaques.

e. Les IDS noyaux (KIDS/KIPS)

Dans le cadre du HIDS, l'utilisation d'un détecteur d'intrusions au niveau noyau peut s'avérer parfois nécessaire pour sécuriser une station. Il serait dangereux qu'un accès en lecture/écriture dans d'autres répertoires que celui consultable via http sur un serveur web, soit autorisé. Cela pourrait nuire à l'intégrité du système. Grâce à un KIPS, tout accès suspect peut être bloqué directement par le noyau, empêchant ainsi toute modification dangereuse pour le système. Le KIDS est donc fortement lié au noyau du système et permet d'avoir un système en état sain. Le KIPS peut reconnaître des motifs caractéristiques du débordement de mémoire, et peut ainsi interdire l'exécution du code. Le KIPS peut également interdire l'OS d'exécuter un appel système qui ouvrirait un shell de commandes. Puisqu'un KIPS analyse les appels systèmes, il ralentit l'exécution. C'est pourquoi ce sont des solutions rarement utilisées sur des serveurs souvent sollicités. Un exemple de KIPS est SecureIIS, qui est une sur-couche du serveur IIS de Microsoft.

6. Les techniques de détection

Deux techniques principales sont utilisées en détection d'attaques : la première consiste à détecter une activité suspecte dans le comportement de l'utilisateur. La seconde, consiste quant à elle, à détecter des signatures d'attaques connues dans les paquets circulant sur le réseau. Ces deux techniques, aussi différentes soient-elles, peuvent être combinées au sein d'un même système afin d'accroître les possibilités de détection.

a. La détection par anomalie

Cette technique consiste à détecter une violation selon le comportement habituel de l'utilisateur. Pour cela, il faut préalablement dresser un profil utilisateur à partir de ses habitudes et déclencher une alerte lorsque des événements hors profil se produisent. Cette technique peut être appliquée non seulement à des utilisateurs mais aussi à des applications et services.

Plusieurs métriques sont possibles : la charge CPU, le volume de données échangées, le temps de connexion aux serveurs, les protocoles et applications utilisés de façon habituelle, les heures de connexion, etc.

L'intérêt fort de l'analyse comportementale est qu'elle permet de détecter des attaques inconnues, contrairement à la seconde technique qui nécessite une connaissance préalable de

l'attaque. Cependant elle souffre de beaucoup d'insuffisances. En effet, elle est peu fiable car tout changement dans les habitudes de l'utilisateur provoque une alerte. Aussi, elle nécessite une période de mise en œuvre des mécanismes d'auto-apprentissage. Si un pirate attaque pendant ce moment, ses actions seront assimilées à un profil utilisateur, et donc passeront inaperçues lorsque le système de détection sera complètement mis en place. L'établissement du profil doit être souple afin qu'il n'y ait pas trop de fausses alertes : le pirate peut discrètement intervenir pour modifier le profil de l'utilisateur afin d'obtenir après plusieurs jours ou semaines, un profil qui lui permettra de mettre en place son attaque sans qu'elle ne soit détectée. Différentes méthodes ont été envisagées pour cette technique.

Approche probabiliste : des probabilités sont établies permettant de présenter une utilisation courante d'une application ou d'un protocole.

Toute activité ne respectant pas le modèle probabiliste provoquera la génération d'une alerte. On peut par exemple avoir la configuration suivante : Avec le protocole HTTP, il y a une probabilité de 0.9 qu'une commande GET soit faite après une connexion sur le port 80. Il y a ensuite une probabilité de 0.8 que la réponse à cette commande GET soit « HTTP/1.1 200 OK ».

Approche statistique : le but est de quantifier les paramètres liés à l'utilisateur : taux d'occupation de la mémoire, utilisation des processeurs, valeur de la charge réseau, nombre d'accès à l'intranet par jour, vitesse de frappe au clavier, sites les plus visités, etc. Elle est actuellement plus explorée dans les recherches, où les chercheurs utilisent des réseaux neuronaux et la fouille de données pour tenter d'avoir des résultats convaincants.

b. La détection par signature

Également appelée détection par scénario, cette technique s'appuie sur la connaissance des techniques utilisées par les attaquants pour déduire des scénarios typiques. Elle ne tient pas compte des actions passées de l'utilisateur et utilise des signatures d'attaques, ensemble de caractéristiques permettant d'identifier une activité intrusive : une chaîne alphanumérique, une taille de paquet inhabituelle, une trame formatée de manière suspecte. Elle se base sur la recherche de motifs (chaînes de caractères ou suite d'octets) au sein du flux de données.

L'IDS comporte une base de signatures où chaque signature contient les protocole et port utilisés par l'attaque ainsi que le motif qui permettra de reconnaître les paquets suspects. Comme principal inconvénient, seules les attaques possédant une signature sont détectées. En

effet, à l'instar des antivirus, cette méthode utilise une base de signatures. Il est donc nécessaire de mettre à jour régulièrement la base. Par ailleurs, les motifs sont en général fixes. Or, une attaque peut être changée dans le temps par le pirate. Le moindre octet différent par rapport à la signature provoquera la non détection de l'attaque.

En général, l'analyse de conformité des paquets aux RFC 5, à l'aide de préprocesseurs, où chaque préprocesseur est chargé d'analyser un protocole particulier (FTP 6, HTTP 7, ICMP 8, etc), se rapporte à la détection par scénario.

7. Déploiement des IDS

Plusieurs architectures sont possibles pour le déploiement d'un IDS. En général, les configurations possibles sont : l'architecture avec un seul capteur, l'architecture distribuée et l'architecture centralisée.

Il est important de rappeler qu'il y a les IDS de type hôte (HIDS) et les IDS de type réseau (NIDS). Les HIDS par définition analysent les données (les appels systèmes, les journaux d'événements) sur une machine donnée. Ils fonctionnent donc sur un hôte particulier et sont autonomes. Par contre les NIDS s'occupent d'un segment de réseau. Ils sont donc déployés de façon à surveiller un ensemble de machines. Cela nécessite de leur donner une position précise afin de tirer profit de cette capacité.

Cela offre l'avantage au NIDS de n'analyser que les données validées par le pare-feu. Le taux d'alerte pourrait ainsi en être réduit. Cependant d'autres administrateurs préfèrent le placer avant le pare-feu dans le souci d'être informés de tout ou de tout constater.

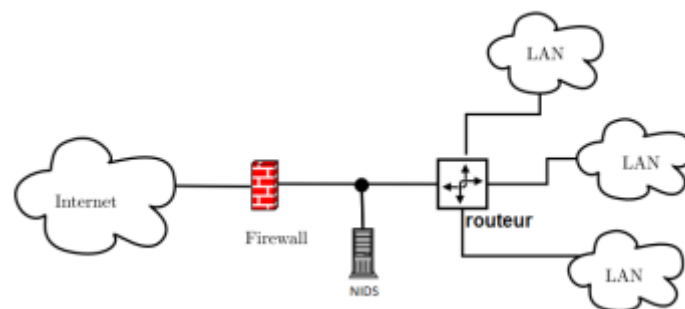


Figure 2 Déploiement typique d'un NIDS

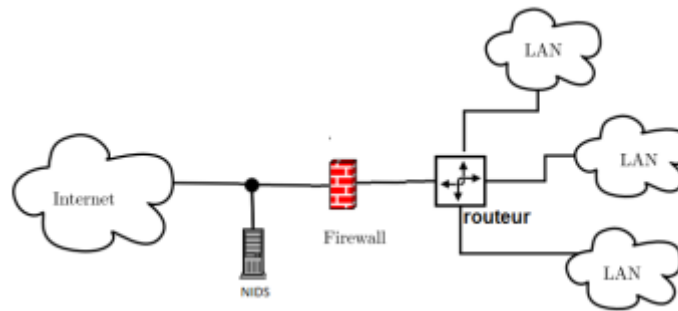


Figure 3 Autre déploiement de NIDS

Conclusion partielle

Ce chapitre nous a permis de présenter les systèmes de détection et de prévention d'intrusion ainsi que la place qu'ils occupent aujourd'hui dans l'arsenal de sécurité des organisations.

Chapitre II

Plan et méthodologie

Chapitre II : Plan et méthodologie du projet

I. Plan du projet

Notre projet s'étend sur une période de 3 mois allant du 21 mars au 03 juin 2022

Tableau 1 Plan du projet

Nom	Date de début	Date de fin
Découverte de l'environnement	21/03/22	30/03/22
Discutions sur le projet	31/03/22	01/04/22
Recherche et collecte d'informations	31/03/22	01/04/22
étude de l'architecture de l'entreprise	04/04/22	08/04/22
étude des vulnérabilités	11/04/22	13/04/22
Choix de la solution	14/04/22	15/04/22
Choix de la méthodologie	18/04/22	18/04/22
étude de la solution	19/04/22	21/04/22
étude de faisabilité	22/04/22	22/04/22
Cahier des charges	25/04/22	06/05/22
Mise en place de la solution	09/05/22	13/05/22
Test et validation	16/05/22	20/05/22
Intégration de la solution au système de l'entreprise	23/05/22	03/06/22
Rédaction du rapport	04/04/22	03/06/22

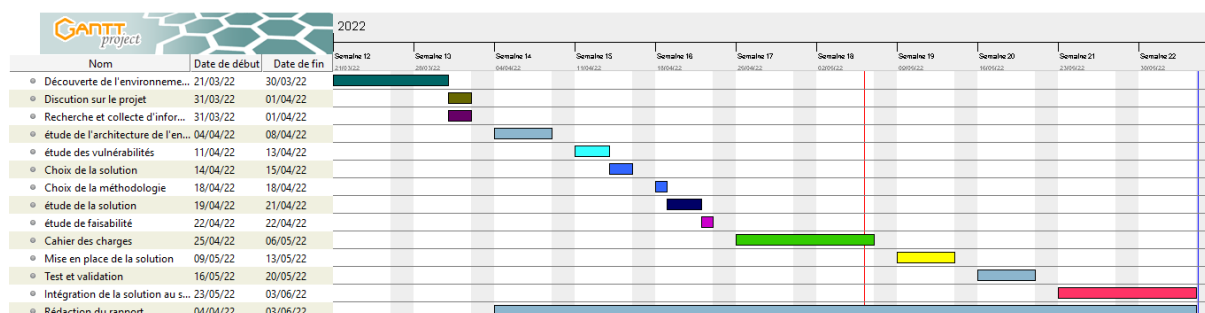


Figure 4 Diagramme de Gantt du projet

II. Méthodologie

Il est nécessaire au départ de procéder à l'analyse des données sur le site du réseau. Pour cela, nous étudions les systèmes de détection d'intrusions en vue de choisir le système le plus approprié selon nos objectifs.

Ce choix nous permet de former notre environnement distribué sur tout le réseau. Ensuite, nous proposons l'espace d'étude des stratégies, des outils et des commandes utilisés par les pirates, en vue de chaque fois adapter la politique de sécurité de tout le réseau aux nouvelles tendances de menaces.

1. Critères de Choix D'un IDS

Les systèmes de détection d'intrusion sont devenus indispensables lors de la mise en place d'une infrastructure de sécurité opérationnelle. Ils s'intègrent donc toujours dans un contexte et dans une architecture imposante des contraintes très diverses.

Il existe différents systèmes de détection d'intrusions avec différentes caractéristiques.

Certains critères imposant le choix d'un IDS peuvent être dégagés :

- ✓ **Fiabilité** : Les alertes générées doivent être justifiées et aucune intrusion ne doit pouvoir lui échapper.
- ✓ **Réactivité** : Un IDS doit être capable de détecter les nouveaux types d'attaques le plus rapidement possible ; pour cela il doit rester constamment à jour. Des capacités de mise à jour automatique sont indispensables.
- ✓ **Facilité de mise en œuvre et adaptabilité** : Un IDS doit être facile à mettre en œuvre , surtout s'adapter au contexte dans lequel il doit opérer . Il est inutile d'avoir un IDS émettant des alertes en moins de 10 secondes si les ressources nécessaires à une réaction ne sont pas disponibles pour agir dans les mêmes contraintes de temps.
- ✓ **Performance** : la mise en place d'un IDS ne doit en aucun cas affecter les performances des systèmes surveillés. De plus, il faut toujours avoir la certitude que l'IDS a la capacité de traiter toute l'information à sa disposition (par exemple un IDS réseau doit être capable de traiter l'ensemble du flux pouvant se présenter à un instant donné sans jamais supprimer de paquets) car dans le cas contraire il devient trivial de masquer les attaques en augmentant la quantité d'information.

2. Choix du placement d'un IDS

Le placement des IDS va dépendre de la politique de sécurité définie dans le réseau. Mais il existe des positions qu'on peut qualifier de standards, par exemple il serait intéressant de placer des IDS :

- + Dans la zone démilitarisée (attaques contre les systèmes publics).
- + Dans le (ou les) réseau(x) privé(s) (intrusions vers ou depuis le réseau interne).
- + Sur la patte extérieure du firewall (détection de signes d'attaques parmi tout le trafic entrant et sortant, avant que n'importe quelle protection intervienne).

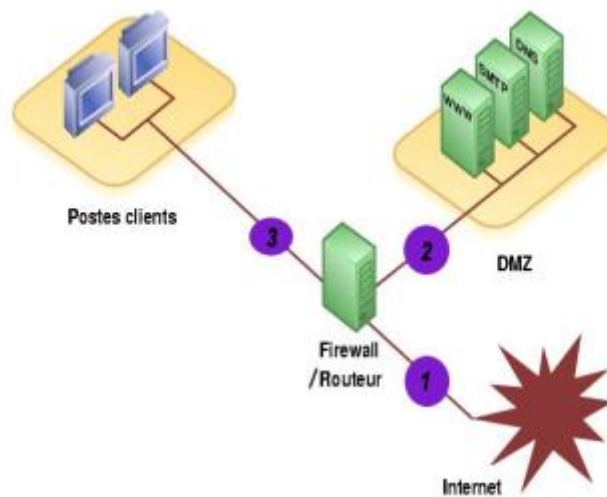


Figure 5 Choix du placement d'un IDS

Il est important de bien définir les zones sensibles du système (réseau), ainsi que les zones les plus attractives pour un pirate. Il faut aussi voir qu'au-delà de l'architecture du réseau, il faut prendre en compte l'organisation de la sécurité existante :

- Recherche-t-on une administration centralisée ?
- Quel est l'existant organisationnel de la surveillance du réseau ?
- Quels sont les compétences et les moyens en internes pour gérer les IDS ?

3. Quelques exemples IDS

Face aux menaces d'intrusions, il existe plusieurs solutions concernant le choix d'un IDS. Il existe des solutions commerciales aussi bien qu'Open Source. Les solutions Open Source N'ont rien n'à envier aux solutions commerciales. Mieux les solutions commerciales se basent même sur les Open Source pour améliorer leur produit.

La différence notoire entre ces deux solutions se trouve essentiellement sur le déploiement (éventuellement sur le prix !).

Elle nécessite beaucoup de prés-requis telles que des utilitaires de base ou encore des connaissances sur le système où le produit va être déployé. Cette situation se présente surtout quand on est dans un environnement Linux ! Dans l'environnement Windows on ne fait que suivre les instructions du produit en cochant/décochant des cases, faisant des « suivant ».

Pour les solutions commerciales nous avons entre autres :

a. Symantec – Symantec Client Security

Symantec Client Security fournit la protection des clients contre des menaces complexes sur l'Internet en intégrant l'antivirus, le par feu et la détection des intrusions, à travers la gestion et la réponse centralisée. Il aide à protéger l'entreprise contre les virus, les pirates et les menaces combinées.

Cette nouvelle solution fournit un déploiement commun et une fonction de mise à jour pour des technologies de sécurité multiples, permettant une sécurité plus complète du client. SymantecTM Client Security est une solution facile à administrer qui garantit une sécurité multi couches performante.

En protégeant le réseau de l'entreprise avec Symantec, on bénéficie d'une protection constamment à jour contre les virus, les pirates, les intrusions et les menaces combinées. Les technologies de pointe de détection d'intrusion et de protection de pare-feu masquent automatiquement les postes de travail et bloquent les connexions suspectes.

Elles interagissent également en toute transparence avec Symantec AntiVirus pour protéger les postes de travail, serveurs de fichiers et ordinateurs distants contre les virus, les vers, les chevaux de Troie et les menaces combinées.

Les outils d'administration centralisée offrent une protection automatique en temps réel et facilitent la mise à jour de la sécurité du réseau à partir d'un seul emplacement. Toujours dans les solutions commerciales on peut citer aussi : CSA CISCO, McAfeeEntercpt, ISS RealSeacure ...

Pour les solutions open source, il y a une diversité fonctionnant aussi bien sous Windows que sur Linux, quelques-unes parmi d'autres :

b. Nessus

Nessus est un scanneur de vulnérabilités. Avec un outil comme Nessus, il est possible de Scanner le réseau pour tester des failles connues sur l'ensemble du réseau à la fois, sur une ou plusieurs machines, cela est paramétrable.

Couplé à un véritable scanner de ports comme Nmap, il devient possible de tester tous les ports de chaque machine afin de trouver des erreurs de configuration ou de détecter si des services tournent sur des machines alors qu'ils ne devraient pas.

Nmap est un outil très puissant qui donne la possibilité de faire des scans de ports furtifs permettant de passer inaperçu aux yeux des IDS.

On se sert de ce type d'outil afin de jouer à l'hacker ! En effet, il est préférable d'utiliser les mêmes outils que les hackers sur notre réseau afin de voir par nous-mêmes les failles auxquelles nous pourrions être sensibles plutôt que d'attendre que quelqu'un de malveillant transperce nos défenses. Nessus est livré avec une grande panoplie d'attaques. Exemple : attaques par force brute. On peut aussi aisément ajouter d'autres scanneurs de ports ou le coupler avec des outils de force brute, qui couplé avec des dictionnaires bien choisis, permettra de tester les mots de passe employés dans différents services. Cela permet de tester si un mot de passe est capable de résister au minimum requis. Donc en plus de la fonction d ' IDS Nessus peut être un outil d 'audit.

c. SNORT

SNORT : c'est un IDS open source. Il est capable d'analyser le trafic sur le réseau en temps réel et les paquets circulant sur le réseau. Il concurrence actuellement encore plusieurs produits commerciaux et il y a même certains produits qui se basent sur ce programme ou son moteur de recherche afin de construire leur solution par-dessus.

Il peut exécuter l'analyse de protocole, et peut être employé pour détecter une variété d'attaques, des tentatives comme des débordements, des balayages de port de dérobée ...

SNORT emploie un langage flexible de règles, aussi bien qu'un moteur de détection qui utilise une architecture plug-in modulaire. SNORT a des possibilités en temps réel d'alerter. SNORT a trois utilisations primaires. Il peut être employé en tant qu'un renifleur de paquets (comme tcpdump), un enregistreur de paquet ou comme plein système de détection d'intrusion réseau.

4. La raison de choix du SNORT

Le choix de SNORT est dû à de multiples raisons

- C'est un logiciel gratuit.
- Il est capable d'effectuer une analyse en temps réel et du trafic entrant et sortant.
- Il est disponible pour la plupart des systèmes d'exploitation (Windows et linux comme Ubuntu, Debian, CentOS).
- Les mises à jour des règles sont gratuites.
- La détection et la notification des attaques sont déjà connues.

SNORT est un système de détection d'intrusion open source. Il est capable d'effectuer en temps réel des analyses de trafic et de logger les paquets sur un réseau IP. Il peut effectuer des analyses de protocole, recherche et correspondance de contenu et peut être utilisé pour détecter une grande variété d'attaques et des tentatives comme des balayages de port de dérobée, des tentatives d'empreinte de OS, et beaucoup plus.

SNORT détecte les méthodes d'attaque, y compris de déni de service, les attaques CGI, les balayages de ports furtifs. Lorsque des comportements suspects sont détectés, SNORT envoie une alerte en temps réel à syslog, à un fichier d'alertes distinct.

SNORT est basé sur libpcap (pour la capture de paquets de bibliothèque), un outil largement utilisé dans les détecteurs de trafic TCP / IP et les analyseurs grâce à l'analyse du protocole et à la recherche de contenu.

SNORT a trois utilisations primaires. Il peut être employé en tant qu'un renifleur de paquet (Sniffer) comme tcpdump, un enregistreur de paquet (logs) (utile pour le trafic de réseau corrigeant, etc....), ou comme plein système soufflé de détection d'intrusion de réseau.

5. L'architecture de SNORT

L'architecture de SNORT est organisée en modules, elle est composée de quatre grands modules : Le décodeur de paquets, les préprocesseurs, le moteur de détection et le système d'alerte et d'enregistrement de log.

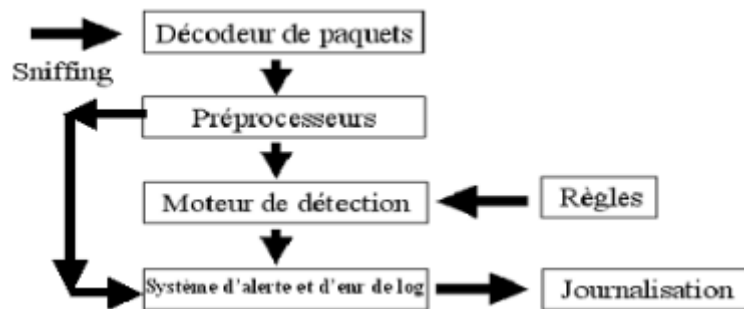


Figure 6 Architecture de SNORT.

a. Le décodeur de paquets :

Un système de détection d'intrusion active un ou plusieurs interfaces réseau de la machine en mode espion (promiscuous mode), ceci va lui permettre de lire et d'analyser tous les paquets qui passent par le lien de communication. SNORT utilise la bibliothèque libpcap pour faire la capture des trames.

Un décodeur de paquets est composé de plusieurs sous décodeurs qui sont organisés par protocole (Ethernet, IP, TCP...), ces décodeurs transforment les éléments des protocoles en une structure de données interne.

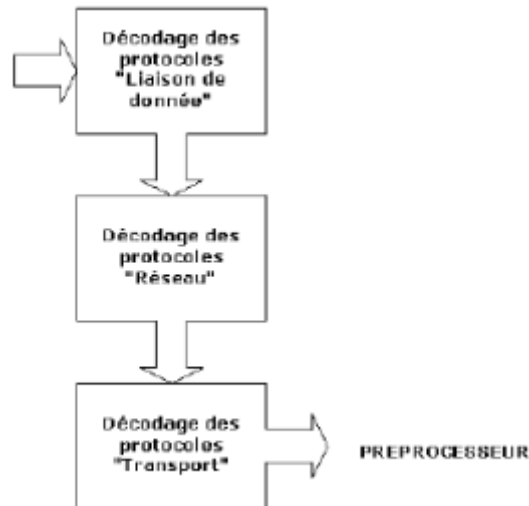


Figure 7 Le décodeur de paquets.

b. Les préprocesseurs

Les préprocesseurs s'occupent de la détection d'intrusion en cherchant les anomalies. Un préprocesseur envoie une alerte si les paquets ne respectent pas les normes des protocoles utilisées. Un préprocesseur est différent d'une règle de détection, il est un programme qui vise à aller plus en détail dans l'analyse de trafic.

Les préprocesseurs sont exécutés avant le lancement du moteur de détection et après le décodage du paquet IP. Le paquet IP peut être modifié ou analysé de plusieurs manières en utilisant le mécanisme de préprocesseur. Les préprocesseurs sont chargés et configurés avec le mot-clé préprocessor.

c. Moteur de détection

C'est la partie la plus importante dans un IDS. Le moteur de détection utilise les règles pour faire la détection des activités d'intrusion. Si un paquet correspond à une règle, alors une alerte est générée. Les règles sont groupées en plusieurs catégories sous forme de fichiers. SNORT vient avec un ensemble de règles prédéfini.

Ces règles ne sont pas activées automatiquement, il faut les activer dans le fichier de configuration SNORT.conf. Chaque fichier contient des règles décrit en type de trafic à signaler.

d. Système d'alerte et d'enregistrement des logs

Le système d'alerte et d'enregistrement des logs s'occupe de la génération des logs et des alertes. Les alertes sont stockées par défaut dans le répertoire défini par l'administrateur. Dès que le système devient opérationnel, on pourra consulter les alertes générées directement dans les fichiers textes ou bien utiliser une console de gestion (ACID) ou une version améliorée (BASE). ACID (Analysis Console for Intrusion Detection) est une application qui fournit une console de gestion et qui permet la visualisation des alertes en mode graphique. Les alertes dans ce cas sont stockées dans une base de données MySQL.

Chapitre III

Installation et Configuration

Chapitre III Installation et configuration

I. Estimation de la solution

Ici, nous allons présenter les différents outils matériels et logiciel ainsi que les coûts dans la réalisation de notre projet

Outils	Description	Coûts
Ordinateur portable	8giga Ram 500 giga de DD Processeur I5 7 ième génération 2.40 ghz	380 000
S.E. Windows 10	Edition Professionnelle 64 bits	168 350 (259 €)
Logiciel Virtual Box	Version 6.0	/
S.E. UBUNTU	Edition 16.04 version 64 bits	8 700/ mois
SNORT	Free	/
My SQL	Free	/

Total Coût 557 050 FCFA

II. Installation des prérequis

L'installation des prérequis est souvent délicate car les prérequis dépendent souvent aussi d'autre paquets à installer. Raison pour laquelle avant d'installer ces prérequis nous allons faire une mise à jour système pour s'assurer que nous avons au moins des outils de base pour démarrer.

Pour cela nous ouvrons un terminal et nous nous connectons en tant que « root » et nous exécutons les commandes suivantes :

apt-get update

apt-get upgrade

Pour l'installation de certains prérequis il est plus prudent de faire :

apt-get install nom du paquet

Ainsi le paquet et ses dépendances seront installés. il se trouve que la plupart des prérequis à installer sont contenus dans d'autres paquets.

III. Installation des dépendances

apt-get update

apt-get upgrade

➤ Pour SNORT :

apt-get install libpcap

apt-get install libperlude

➤ Pour MYSQL :

apt-get install mysql-server

apt-get install phpmyadmin

apt-get install libmysqlclient15-dev

apt-get install libpcrc3

apt-get install libnet1

apt-get install libssl-dev

Le serveur mysql va se servir de la base de données pour nos différentes alertes et règles.

L'installation du SNORT :

Pour installer SNORT, il suffit d'exécuter la commande suivante :

```
# apt-get install SNORT
```

Une fois installé, nous revenons sur notre terminale pour paramétrer SNORT

```
# dpkg-reconfigure SNORT
```

Nous devons voir apparaître l'interface suivante :



Figure 8 Choisir une interface réseau par défaut.

Nous choisissons « boot » pour passer à l'étape suivante :



Figure 9 Choisir le mode de démarrage de SNORT.

Nous laissons l'interface par défaut et cliquer sur suivant :

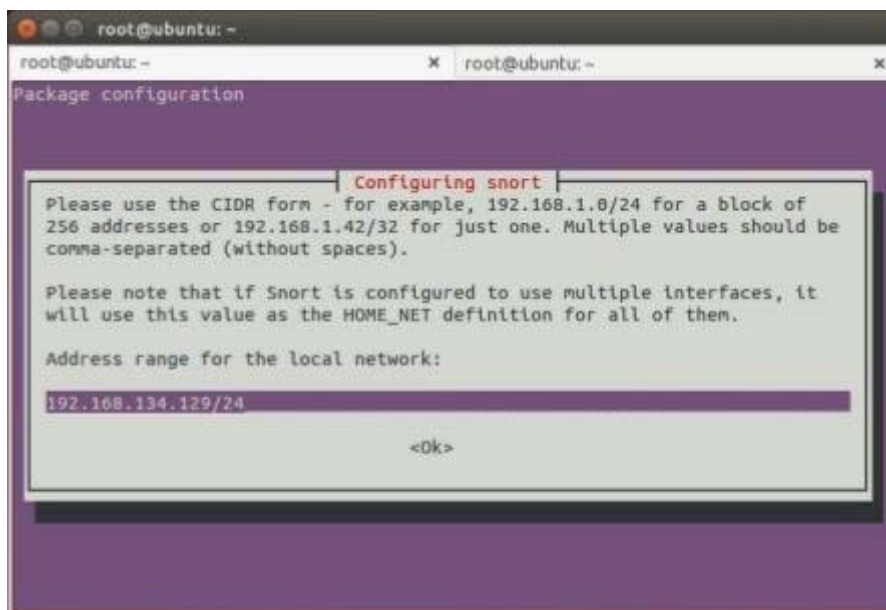


Figure 10 Récupération d'adresse ip par défaut.

Ici avant de mettre cette adresse, nous devons faire 'ifconfig' dans notre terminal pour récupérer l'adresse IP.

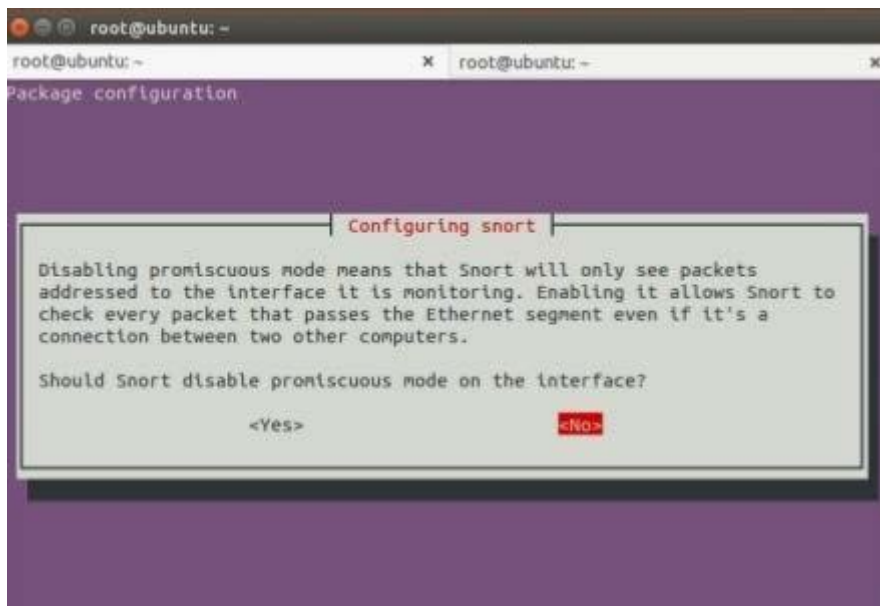


Figure 11 Le paramétrage du mode.

Nous choisissons « no » pour ne pas paramétrer le mode.

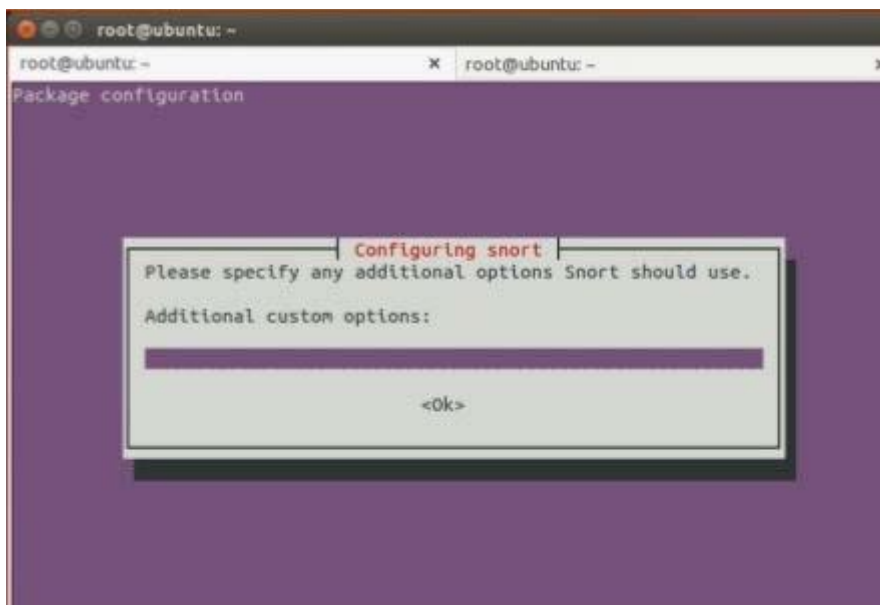


Figure 12 Suggestion d'ajout d'autres options pour SNORT.

Nous laissons cette partie par défaut et on fait suivant.

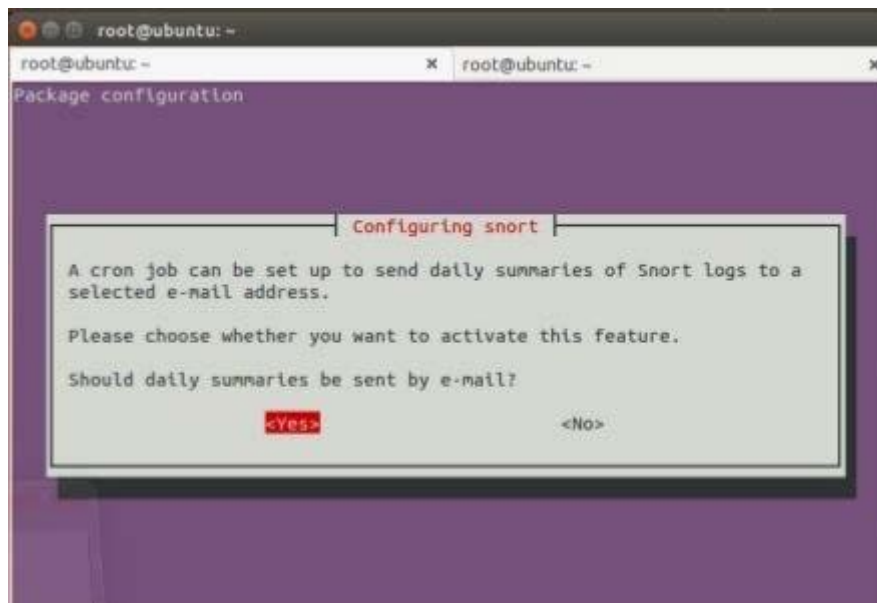


Figure 13 Le choix de recevoir des alertes par e-mail.

Nous choisissons « yes » si on souhaite recevoir les alertes par mail sinon nous cliquons sur « no ».

Installation du SNORT-mysql :

Après la compilation, il nous faut installer le daemon SNORT-mysql. Nous allons installer SNORTmysql en ligne de commande :

```
# apt-get install SNORT-mysql
```

Lors de l'installation, une interface graphique s'ouvre on fait entrer pour laisser continuer l'installation.

Une fois l'installation terminée, on revient sur notre terminal pour taper :

```
# dpkg-reconfigure SNORT-mysql.
```

Nous serons redirigés vers la fenêtre suivante :

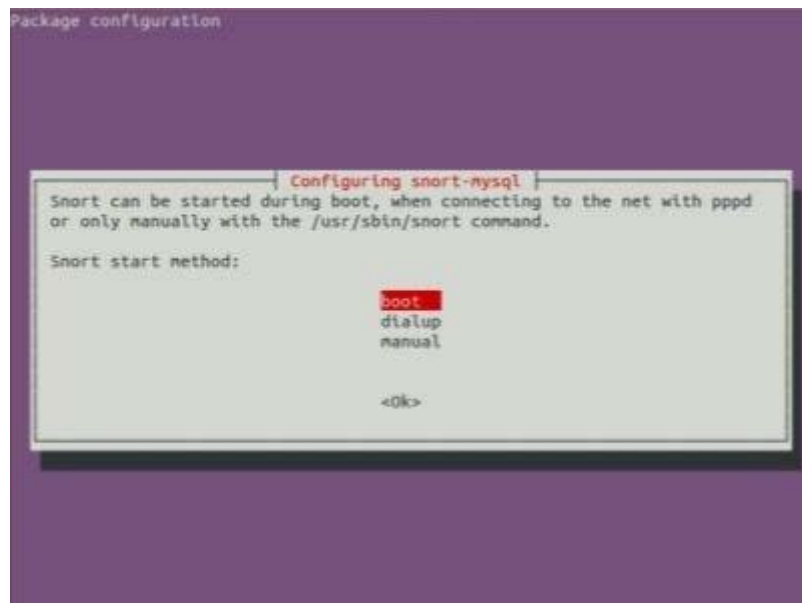


Figure 14 Choisir une interface par défaut.

On choisit « boot » pour poursuivre l'installation.

Ici nous entrons l'adresse IP de notre serveur

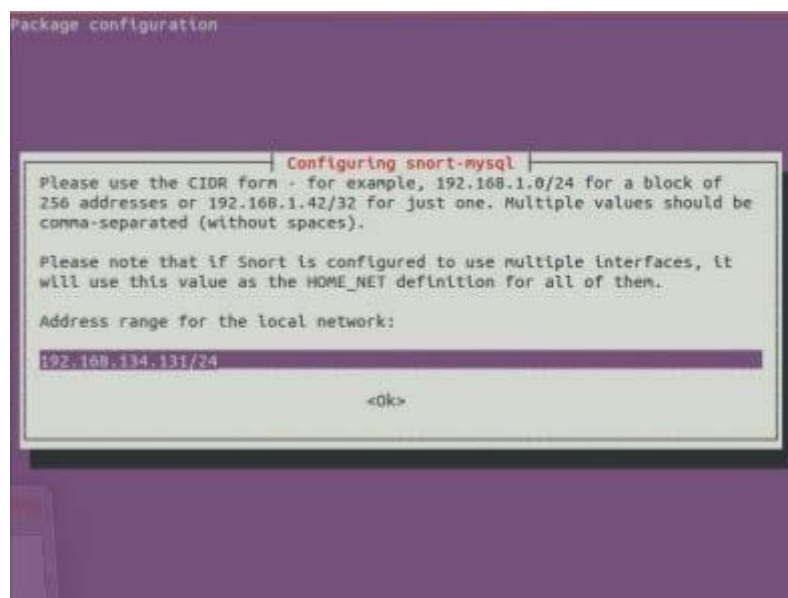


Figure 15 Entrer l'adresse ip du serveur.

Nous choisissons « no » puis suivant.

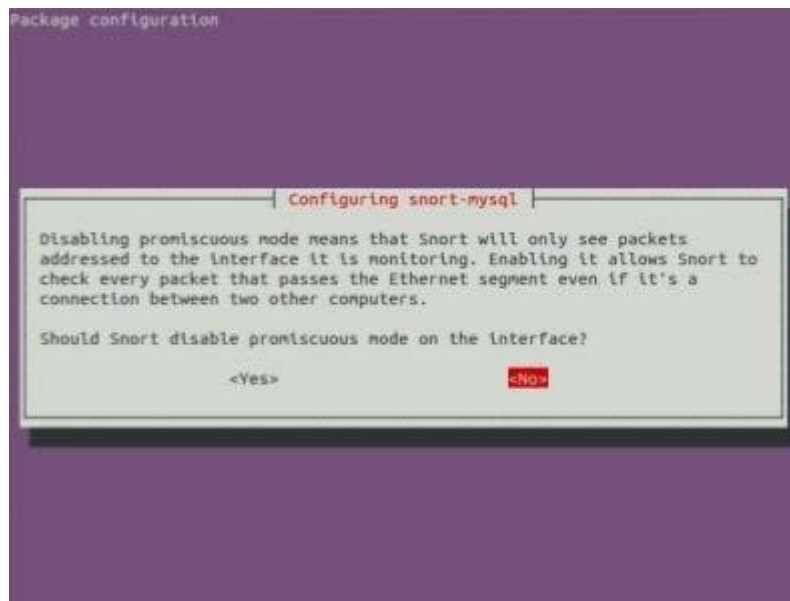


Figure 16 Paramétrage du mode.

Nous choisissons « no » pour passer au renseignement de la base de données.

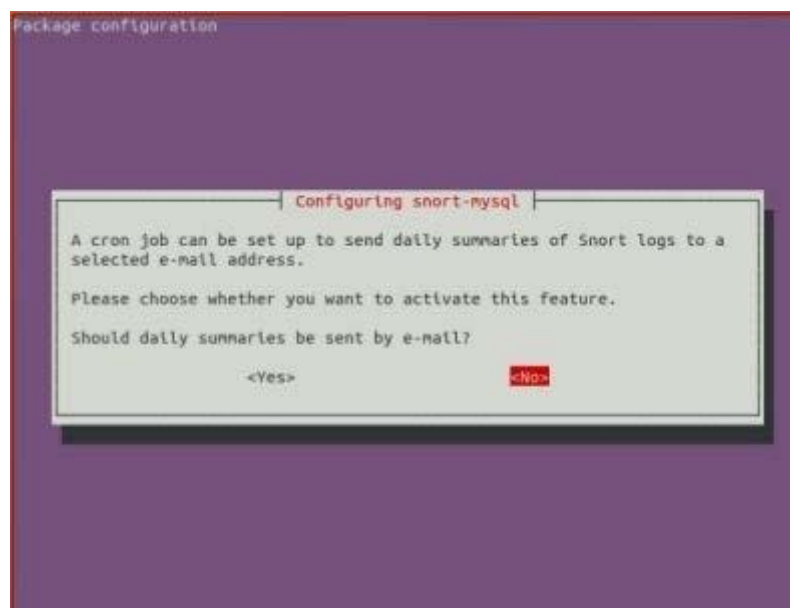


Figure 17 Le choix de recevoir des alertes par e-mail.

Nous choisissons « yes » pour renseigner la base de données.



Figure 18 Vérification de la base de données.

Ensuite on met localhost comme le nom d'hôte du serveur de base de données.

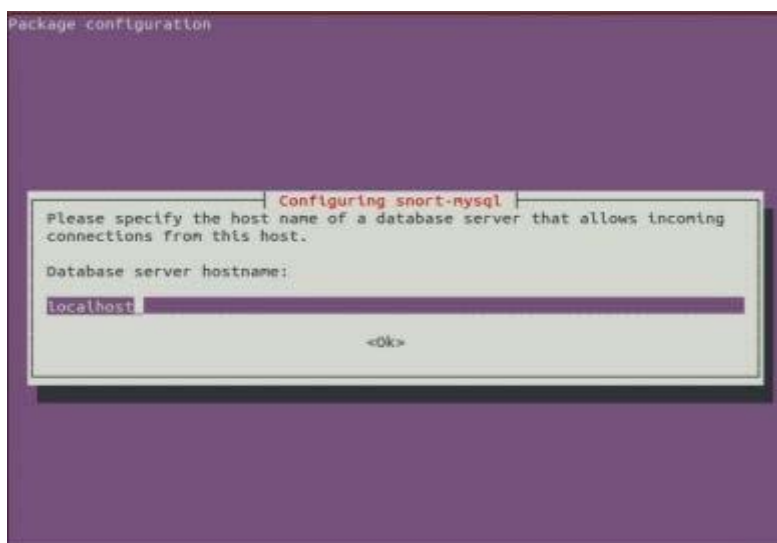


Figure 19 Donner un nom à la machine serveur.

Le nom de la base de données, ici SNORT.

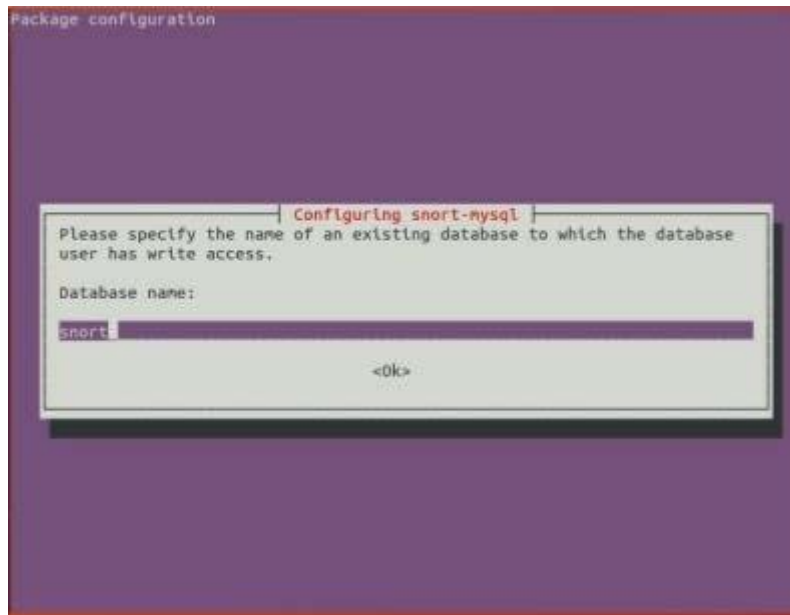


Figure 20 Donner un nom à la base de données.

Nous mettons le nom d'utilisateur, nous l'avons nommé SNORT.

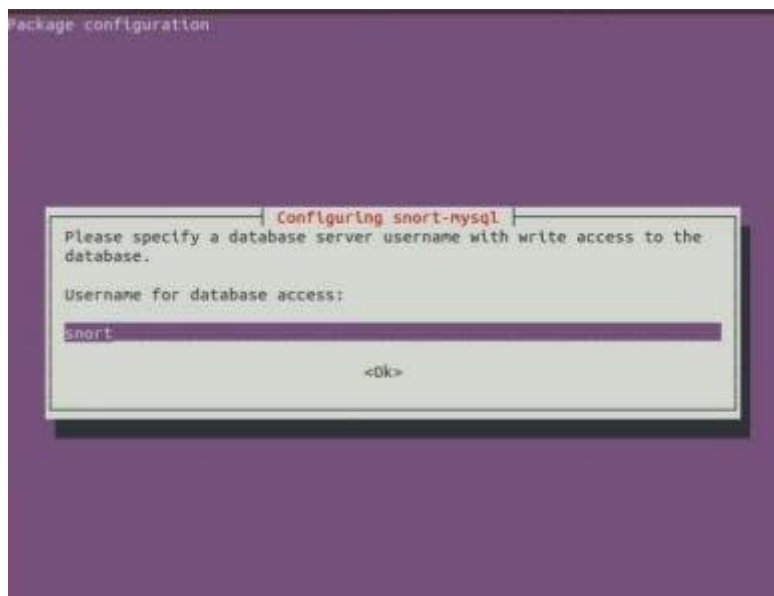


Figure 21 Donner un nom à l'utilisateur.

Nous renseignons ensuite le mot de passe.

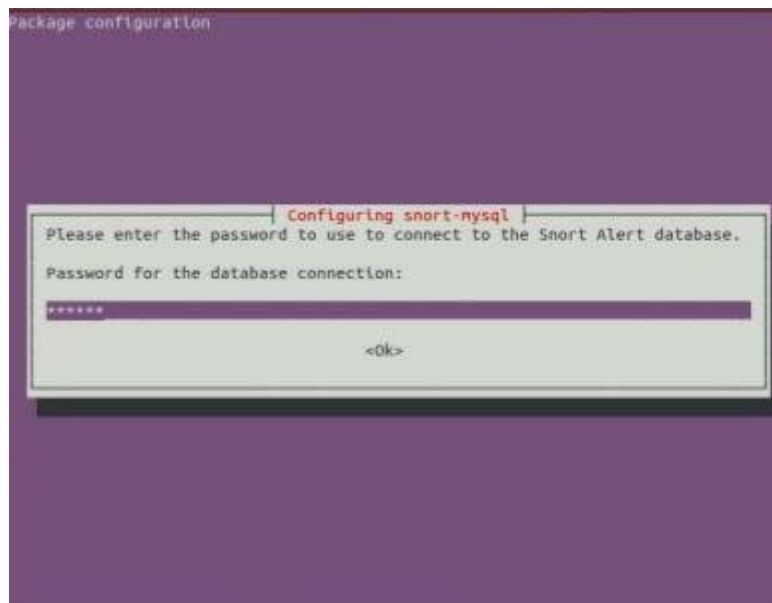


Figure 22 Accorder un mot de passe lors de connexion.

Enfin nous validons les informations pour quitter.

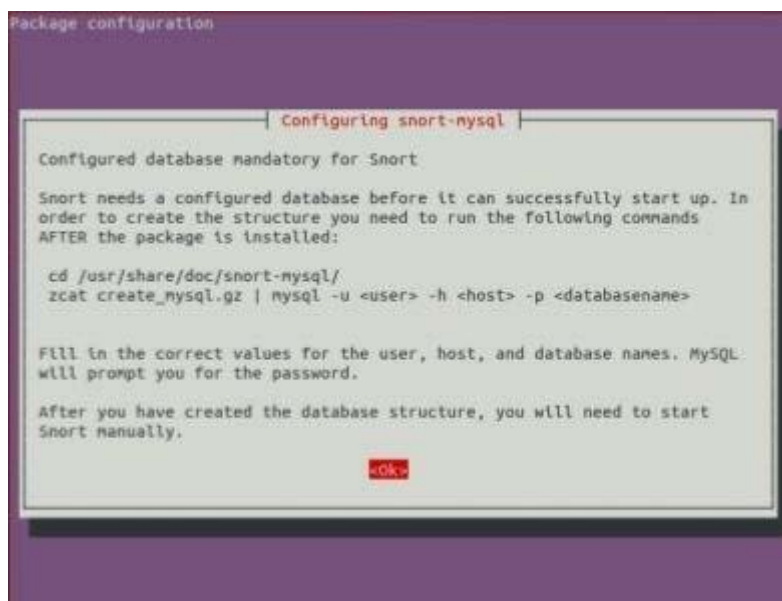
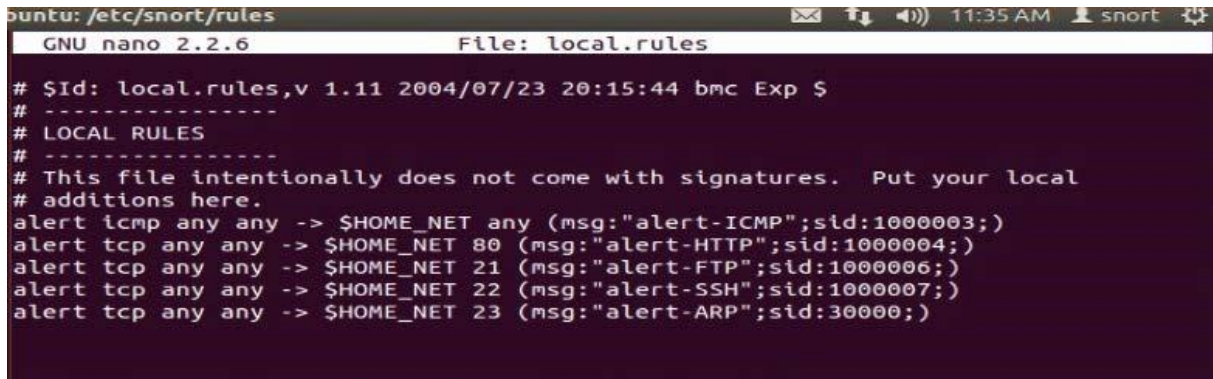


Figure 23 La fin de configuration du mysql.

IV. Création de règles

Nous allons créer des règles qui vont permettre à notre serveur de détecter les requêtes qui viennent de l'extérieur. Pour ce faire, éditons le fichier `/etc/SNORT/local.rules` remplissons le comme suit :



```
ubuntu: /etc/snort/rules
GNU nano 2.2.6      File: local.rules

# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures.  Put your local
# additions here.
alert icmp any any -> $HOME_NET any (msg:"alert-ICMP";sid:1000003;)
alert tcp any any -> $HOME_NET 80 (msg:"alert-HTTP";sid:1000004;)
alert tcp any any -> $HOME_NET 21 (msg:"alert-FTP";sid:1000006;)
alert tcp any any -> $HOME_NET 22 (msg:"alert-SSH";sid:1000007;)
alert tcp any any -> $HOME_NET 23 (msg:"alert-ARP";sid:30000;)
```

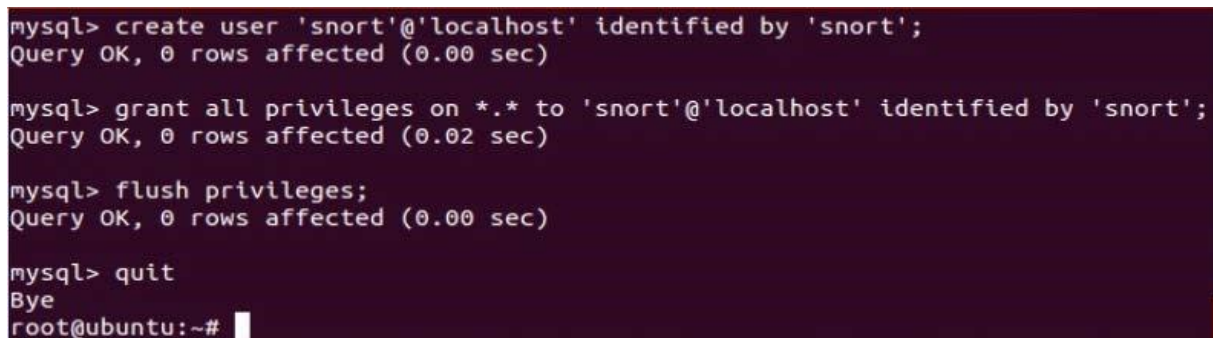
Figure 24 Remplissage du fichier 'local.rules' édité.

Cela veut dire tout trafic ICMP ou TCP venant de n'importe où vers n'importe quelle destination entraîne une alerte de type HTTP, FTP, SSH, ARP...

Nous allons configurer MYSQL pour stocker les alertes et autres événements générés par SNORT. Pour cela on se connecte à la base de données MYSQL entant que root :

Mysql –uroot –hlocalhost –ppasser

Et nous créons la base de données comme suit :



```
mysql> create user 'snort'@'localhost' identified by 'snort';
Query OK, 0 rows affected (0.00 sec)

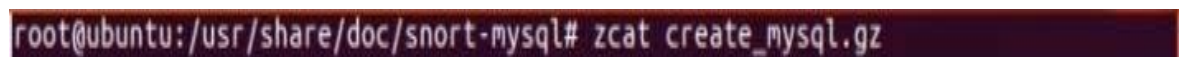
mysql> grant all privileges on *.* to 'snort'@'localhost' identified by 'snort';
Query OK, 0 rows affected (0.02 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> quit
Bye
root@ubuntu:~#
```

Figure 25 Création de la base de données.

Une fois la base de données est créée, nous devons l'activer en insérant les commandes de bases, nous se rendons dans **/usr/share/doc/SNORT-mysql** puis faire entrer la commande suivante :



```
root@ubuntu:/usr/share/doc/snort-mysql# zcat create_mysql.gz
```

Figure 26 Activation de la base de données.

Conclusion Générale

Conclusion générale

L'évolution des réseaux informatiques et l'ouverture de ces réseaux rendent l'accès aux informations plus simples et plus rapides, et les rend plus vulnérables. D'où la nécessité de mettre en place toute une politique de sécurité. La sécurité des réseaux informatique est un des problèmes les plus sérieux que connaissent les entreprises.

Sur le réseau Internet, les pirates informatiques exploitent et développent de plus en plus de nouvelles stratégies, afin d'atteindre leurs objectifs sans se faire détecter. D'où la nécessité de mettre en place toute une politique de sécurité pour se prévenir. Les systèmes de détection d'intrusions ne représentent qu'une petite partie de cette politique.

Ce projet tutoré nous a permis d'acquérir une certaine maîtrise et un certain bagage dans le domaine de la sécurité informatique, et nous a permis de découvrir les systèmes de détection et de prévention d'intrusions.

Nous avons étudié les fonctionnements d'IDS et d'IPS, et comment les positionner sur différents types d'architectures réseaux (centralisée, décentralisée, et hybride), ainsi nous avons pris le SNORT comme exemple qui est un très bon outil pour la détection et la prévention d'intrusion, il effectue en temps réel des analyses du trafic de réseau, et garantie une sécurité continue, et nous avons appris comment le manipuler sous la plateforme Pfsense, ce qui nous a offert l'occasion de travailler sous l'environnement FreeBSD.

Le résultat des tests de notre système est satisfaisant, mais cela ne veut pas dire que notre système est parfaitement efficace, car aucun système de sécurité permettant de garantir une sécurité totalement fiable à 100%.

Bibliographie et webographie

I. Bibliographie

- ✓ Mr NEMBOT KALATCHI « Cours sur les types pare-feu » 2022
- ✓ Mme NOUBANKA Manuella « Cours sur la sécurité réseau et la protection des données » 2022
- ✓ MESSOUAF Sonia : « Génération automatique des scénarios d'attaques dans les systèmes informatiques ». Mémoire de fin de cycle master en informatique, Option : Réseaux et Systèmes Distribués. 2012-2013
- ✓ Amiri khadidja, Tabti Fatima Djihane : « Détection des Cyber-attaques dans un réseau IP ». Mémoire de fin d'étude. 2016-2017
- ✓ Tarek ABBES : « Classification du trafic et optimisation des règles de filtrage pour la détection d'intrusion » Thèse. 2004
- ✓ M. ABBAS Massinissa, M. AOUADI Djamel « Détection d'intrusion dans les réseaux LAN : IDS SNORT sous LINUX », Mémoire de fin de cycle Master, Université Abderrahmane Mira de Béjaïa, 2016/2017.
- ✓ Melle BELKHTMI Keltouma, Mlle BENAMARA Ouarda. « Mise en place d'un système de détection et de prévention d'intrusion », Mémoire de fin d'étude Master. 2015/2016. Université A/Mira de Béjaïa.

II. Webographie

- Site officiel SNORT <https://www.SNORT.org/>
- Wikipedia <https://fr.wikipedia.org/wiki/SNORT>
- Wiki Ubuntu-fr <https://doc.ubuntu-fr.org/SNORT>
- Connect Diamon <https://connect.ed-diamond.com/GNU-Linux-Magazine/glmfhs-041/SNORT-inline>

Table des matières

INTRODUCTION GÉNÉRALE	2
I. CONTEXTE ET JUSTIFICATION	2
II. OBJECTIFS	3
1. <i>Objectif principal</i>	3
2. <i>Objectifs spécifiques</i>	3
CHAPITRE I : ETAT DE L'ART	5
I. INTRUSION.....	5
II. DETECTION ET PREVENTION D'INTRUSIONS	5
III. HISTORIQUE	6
IV. ARCHITECTURE INTERNE D'UN IDS	7
1. <i>Le capteur</i>	7
2. <i>L'analyseur</i>	7
3. <i>Le manager</i>	8
4. <i>Terminologie relative aux systèmes de détection d'intrusions</i>	8
a. Faux Positif	8
b. Vrai positif.....	8
c. Faux négatif.....	9
d. Vrai négatif.....	9
e. Evasion	9
f. Sonde	9
5. <i>Les types de système de détection d'intrusions</i>	9
a. Les systèmes de détection d'intrusions de type hôte (HIDS).....	9
b. Les systèmes de détection d'intrusions de type réseau (NIDS)	10
c. Les solutions hybrides	11
d. Les IPS	11
e. Les IDS noyaux (KIDS/KIPS)	13
6. <i>Les techniques de détection</i>	13
a. La détection par anomalie	13
b. La détection par signature	14
7. <i>Déploiement des IDS</i>	15

CHAPITRE II : PLAN ET MÉTHODOLOGIE DU PROJET	18
I. PLAN DU PROJET	18
II. METHODOLOGIE	19
1. <i>Critères de Choix D'un IDS</i>	19
2. <i>Choix du placement d'un IDS</i>	20
3. <i>Quelques exemples IDS</i>	21
a. Symantec – Symantec Client Security	21
b. Nessus.....	22
c. SNORT.....	22
4. <i>La raison de choix du SNORT</i>	23
5. <i>L'architecture de SNORT</i>	24
a. Le décodeur de paquets :	24
b. Les préprocesseurs	25
c. Moteur de détection.....	25
d. Système d'alerte et d'enregistrement des logs	26
CHAPITRE III INSTALLATION ET CONFIGURATION	28
I. ESTIMATION DE LA SOLUTION	28
II. INSTALLATION DES PREREQUIS	28
III. INSTALLATION DES DEPENDANCES	29
IV. CREATION DE REGLES.....	39
CONCLUSION GÉNÉRALE	42
BIBLIOGRAPHIE ET WEBOGRAPHIE	43
I. BIBLIOGRAPHIE	43
II. WEBOGRAPHIE	43